M3Time: LLM-Enhanced Multi-Modal, Multi-Scale, and Multi-Frequency Multivariate Time Series Forecasting

Shuning Jia[1,2*], **Baijun Song**[1*], **Canming Ye**[1], **Chun Yuan**[1†]
[1]Tsinghua University
[2]Shenzhen University
shuningjia524@gmail.com, yuanc@sz.tsinghua.edu.cn

Abstract

Multivariate Time Series Forecasting (MTSF) aims to capture the dependencies among multiple variables and their temporal dynamics to predict future values. In recent years, Large Language Models (LLMs) have set a new paradigm for MTSF, incorporating external knowledge into the modeling process through textual prompts. However, we observe that current LLM-based methods fail to exploit these priors due to their coarse-grained representation of time series data, which hinders effective alignment of the two modals. To address this, we propose M3Time, a multi-modal, multi-scale, and multi-frequency framework for multivariate time series forecasting. It enhances the quality of time series representations and facilitates the integration of LLM semantic priors with fine-grained temporal features. Additionally, M3Time further improved training stability and model robustness with an adaptive mixed loss function, which dynamically balances L1 and L2 error terms. Experiment results on seven real-world public datasets show that M3Time consistently outperforms state-of-the-art methods, underscoring its effectiveness.

Introduction

Time series forecasting has long been a widely studied field, with broad applications such as traffic prediction (Liu et al. 2024a), power load forecasting (Jalalifar, Delavar, and Ghaderi 2024), and weather prediction (Wu et al. 2021; Bi et al. 2023). Given its significant importance in the real world, MTSF has gradually become a critical research frontier.

Deep neural networks have revolutionized time series forecasting in recent years, evolving from early recurrent neural networks (RNNs), multi-layer perceptrons (MLPs), convolutional neural networks (CNNs), Transformers, to hybrid architectures, and large pretrained models. Motivated by the success of LLMs in natural language processing (NLP), researchers have been exploring ways of utilizing their semantic prior knowledge for time series forecasting (Jin et al. 2023b). Current methods can be broadly categorized into two groups: (1) Directly feeding or reprogramming numerical sequences into LLMs for prediction (Zhou et al. 2023; Jin et al. 2023a; Jia et al. 2024; Gruver et al. 2023; Cao et al. 2023), which struggle to model complex temporal dependencies due to the lack of structured representations; (2) Incorporating textual prompts to guide side models in understanding the task (Liu et al. 2025; Bumb et al. 2025), which use the semantic representations of LLMs to enhance the numerical representations of other base models for prediction.

Existing studies (Li et al. 2025; Liu et al. 2025; Jin et al. 2023a) have shown that when the base model is relatively weak (Zeng et al. 2023; Nie et al. 2022; Zhou et al. 2021), incorporating textual priors can yield performance improvements, as the latter serves as an auxiliary to the former to compensate for its limited capability. However, does such improvement indicate that the weak models are truly capable of bridging the modality gap and exploiting the semantics of text?

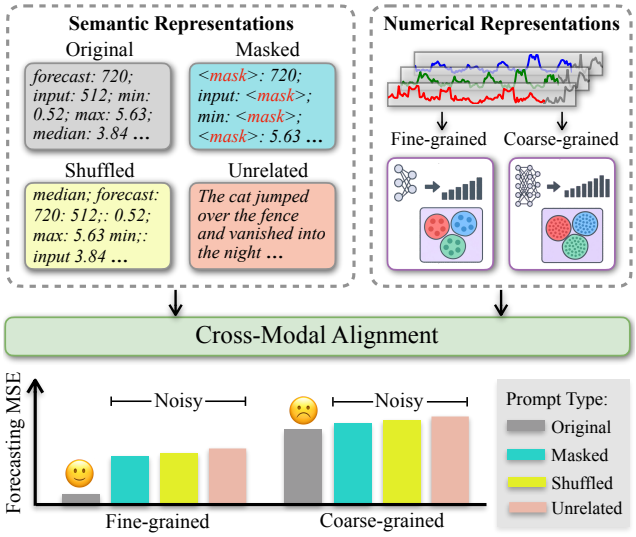

Figure 1: Our exploratory experiments. Fine-grained and coarse-grained numerical representations from base models are aligned with semantical representations from four types of prompts to evaluate the extent to which weak and strong base models can exploit textual prompts.

Through exploratory experiments, as illustrated in Fig-

*These authors contributed equally.
†Corresponding author.

ure 1, we observe that for multi-modal architectures that align numerical embeddings from the base models with semantic embeddings from textual prompts, replacing the original meaningful prompts with noisy interfering prompts (such as masked, shuffled, or unrelated prompts) has minimal adverse effect on their forecasting performance. In contrast, when the base models are much stronger, the forecasting performance improves, and on top of that, this noise has a much worse impact on the results. These findings suggest that the performance improvement observed in weak models does not stem from a deep understanding of textual semantics, but rather from a superficial feature extraction. Therefore, we argue that only when the base models are strong and can generate high-quality, fine-grained numerical representations can they effectively align with the semantic representations of textual prompts, truly utilizing the semantic priors embedded in LLMs.

Based on these insights, we design an end-to-end, LLM-enhanced, multi-modal, multi-scale, multi-frequency, multivariate time series forecasting framework, called **M3Time**. As illustrated in Figure 2, M3Time performs MTSF through the Following core modules: (1) Cross-Variable Interaction captures inter-variable dependencies through self-attention; (2) Multi-Scale Temporal Enhancement performs trend-season decomposition and multi-scale downsampling to enhance representations temporally; (3) Dynamic Convolution Block performs period-driven 2D transformation, applies 2D convolution and aggregation, to capture fine-grained inter and intra-period patterns, improving the quality of representations; (4) Multi-Scale Recursive Aggregation recursively aggregates trends and seasons across multiple scales through cross-attention; (5) LLM Semantic Enhancement uses descriptive prompt as prefix to generate semantic representations, then fuse them into the fine-grained numerical representations via cross-attention, effectively enhancing presentations through semantic guidance.

Additionally, M3Time stabilizes training by adopting an Adaptive Mixed Loss, which dynamically balances itself between L1 and L2 error. We conduct extensive experiments on seven real-world datasets, and M3Time consistently achieves state-of-the-art performance in time series forecasting tasks. Our contributions are summarized as follows:

- Our experiments show that weak time series base models mainly rely on superficial feature extraction when incorporating textual prompts, lacking the ability to understand their semantics. Instead, only strong models that generate fine-grained representations can effectively interpret textual semantics and unlock the full potential of LLMs.
- We propose M3Time, a LLM-enhanced, multi-modal, multi-scale, and multi-frequency time series forecasting framework that enables better prompt alignment through fine-grained temporal modeling.
- We conduct extensive experiments to evaluate M3Time on real-world datasets, showcasing its state-of-the-art forecasting performance.

Related Work

Deep Forecasting Models Time series forecasting has progressed from RNNs that model probabilistic dynamics (Salinas et al. 2020; Shen, Li, and Kwok 2020), which were limited by the Markov assumption (Hochreiter and Schmidhuber 1997), to convolutional encoders that model 1D temporal dependencies (Franceschi, Dieuleveut, and Jaggi 2019), or model 2D periodic patterns (Wu et al. 2022; Nematirad, Pahwa, and Natarajan 2025), or model multi-resolution views and long horizons (Liu et al. 2022). Additionally, minimalist MLP or linear baselines (Zeng et al. 2023; Wang et al. 2024b) then showed that careful trend-season decomposition plus simple scale-mixing can match heavier architectures at a fraction of the compute. Transformer variants now dominate: trend-season decomposition with autocorrelation (Wu et al. 2021), frequency-enhanced decomposition (Zhou et al. 2022), patchified channel-independent tokenization for long contexts (Nie et al. 2022), explicit cross-dimension attention (Zhang and Yan 2023), and variate-as-token inversion to emphasize inter-series structure (Liu et al. 2023). These innovations collectively highlight the importance of (1) decomposing periodic/aperiodic structure, (2) re-indexing or reshaping series by patching or 2D views, and (3) decoupling temporal vs. variable modeling. Inspired by NLP/vision foundation models, the time-series community now pre-trains large Transformers on billions of real-world points to build universal numeric backbones. Decoder-only architectures (Das et al. 2024; Rasul et al. 2023) deliver strong zero-shot forecasts, demonstrating the effectiveness of autoregressive pretraining. On the other hand, encoder-only architectures, with various designs including learnable task tokens (Challu et al. 2023), any-variate masked attention (Woo et al. 2024), and more (Goswami et al. 2024), unify forecasting, imputation, anomaly detection, and classification across heterogeneous domains.

LLM-Enhanced Forecasting Beyond numeric-only pretraining, researchers increasingly tap the broad sequence priors of LLM by aligning time series data with token-based text. Simply encoding numbers as character strings (Gruver et al. 2023) can unlock surprising zero-shot extrapolation in off-the-shelf LLMs, and quantization-to-vocabulary (Ansari et al. 2024) enables probabilistic decoding and rapid transfer at scale. Some approaches adapt pretrained decoders to time-series inputs with minimal parameter updates: a lightly tuned GPT-2 can model channel-structured patches for downstream tasks (Zhou et al. 2023) or be modality-aligned via learned textual prototypes and prompt-as-prefix conditioning (Jin et al. 2023a). Prompt- and in-context methods push further by projecting numeric histories into an LLM embedding space and autoregressively generating multi-step forecasts (Liu et al. 2024b) or by patch-based structured prompts that incorporate decomposition, neighbor retrieval, and instruction cues for zero-shot use (Nie et al. 2022). Recent cross-modality alignment frameworks couple dedicated numeric encoders with language-derived semantic embeddings to reduce modality gaps (Liu et al. 2025), and multi-modal prompting pipelines extend this

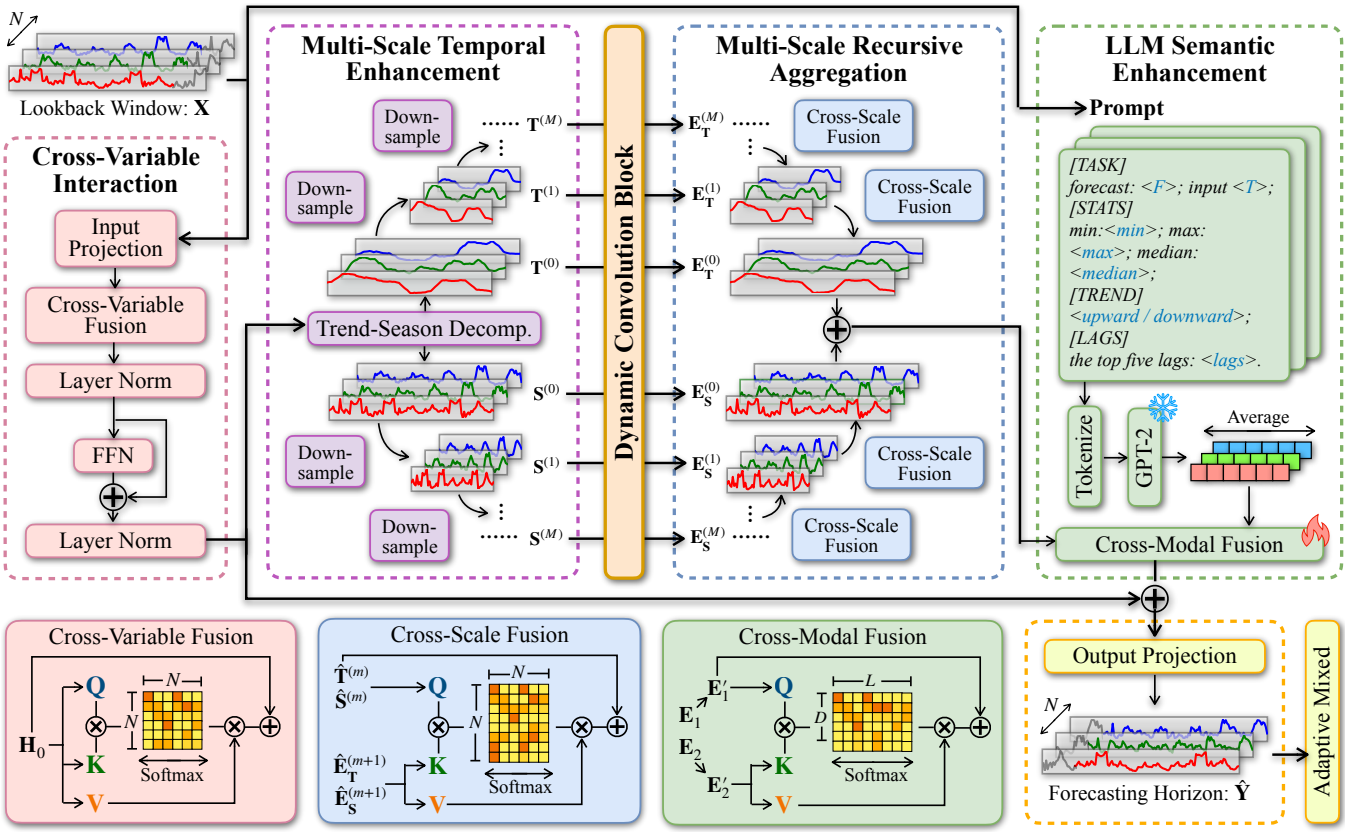

Figure 2: Overview of M3Time.

idea to paired numerical–text corpora (Jia et al. 2024). Together, these studies indicate that, when properly aligned, LLMs can contribute domain-agnostic reasoning and compositional priors that classical numeric models lack, pointing towards a new generation of semantics-aware time-series forecasters.

Method

Cross-Variable Interaction

To effectively capture inter-variable dependencies, we introduce a Cross-Variable Interaction module. Given a multivariate time series input sequence $\mathbf{X}_T \in \mathbb{R}^{N\times T}$, first, we project the input time series sequence into a latent space by linear transformation:

$$\mathbf{H}_0 = \text{Linear}\left(\mathbf{X}_T\right), \tag{1}$$

where $\mathbf{H}_0 \in \mathbb{R}^{N\times D}$ and D denotes the temporal embedding dimension. Then, we regard each variable as a token and feed all N tokens into a Transformer Encoder to obtain variable embeddings:

$$\mathbf{E}_0 = \text{TransformerEncoder}\left(\mathbf{H}_0\right). \tag{2}$$

Multi-Scale Temporal Enhancement

To capture temporal features of multiple time scales, we design a Multi-Scale Temporal Enhancement module.

Trend-Season Decomposition Real-world time series often exhibit distinct trends and seasonal patterns, which differ in their temporal scales and variation behaviors. Therefore, following previous works (Zeng et al. 2023; Zhou et al. 2022; Wang et al. 2023), instead of modeling the raw data directly, we decompose the variable embeddings $\mathbf{E}_0$ into a seasonal component $\mathbf{S}^{(0)}$ and a trend component $\mathbf{T}^{(0)}$:

$$\begin{aligned}\mathbf{T}^{(0)} &= \text{AvgPool1D}\left(\mathbf{E}_0, \text{stride=1}\right),\\ \mathbf{S}^{(0)} &= \mathbf{E}_0 - \mathbf{T}^{(0)},\end{aligned} \tag{3}$$

where $\mathbf{T}^{(0)}, \mathbf{S}^{(0)} \in \mathbb{R}^{N\times D}$.

Multi-Scale Downsampling We downsample the trend and season component along the temporal dimension into M scales of granularity in an exponentially decreasing fashion. Concretely, starting from the original scale $\mathbf{T}^{(0)}$ and $\mathbf{S}^{(0)}$, two adjacent data points are recursively averaged to obtain their next scales:

$$\begin{aligned}\mathbf{T}^{(m)} &= \text{AvgPool1D}\left(\mathbf{T}^{(m-1)}, \text{stride=2}\right),\\ \mathbf{S}^{(m)} &= \text{AvgPool1D}\left(\mathbf{S}^{(m-1)}, \text{stride=2}\right),\end{aligned} \tag{4}$$

where $m \in \{1,\ldots,M\}$, $\mathbf{T}^{(m)}, \mathbf{S}^{(m)} \in \mathbb{R}^{N\times D_m}$, and $D_m = \lfloor \frac{D}{2^m} \rfloor$. This process yields a set of trends $\mathcal{T} = \{\mathbf{T}^{(0)},\ldots,\mathbf{T}^{(M)}\}$ and a set of seasons $\mathcal{S} = \{\mathbf{S}^{(0)},\ldots,\mathbf{S}^{(M)}\}$, each containing $M+1$ scales of granularity. As m increases, the representation becomes more

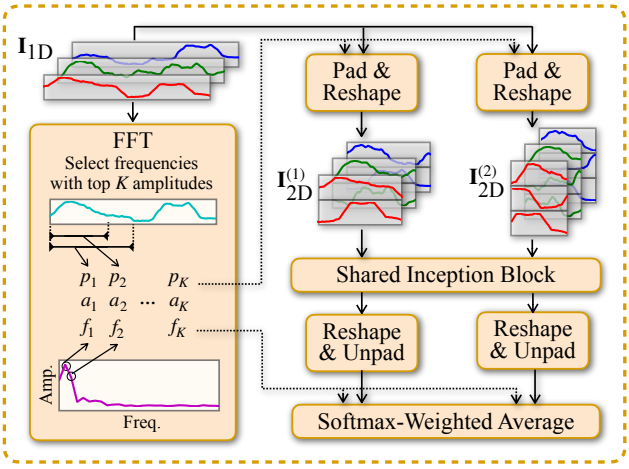

Figure 3: Dynamic Convolution Block.

global. By aggregating information across multiple scales, the model can simultaneously capture both global trends and fine-grained temporal patterns, thereby enhancing its long-term forecasting ability.

Dynamic Convolution Block

The majority of M3Time consists of several Dynamic Convolution Blocks, as illustrated in Figure 3. We design this module to effectively capture inter- and intra-period patterns of multi-scale trends and seasons (Wu et al. 2022). Block index was omitted in this section for simplicity and clarity.

Period-Driven 2D Transformation We identify dominant periods of each component and reshape them for 2D convolution, thereby capturing both inter- and intra-period patterns. Specifically, for each 1D component $\mathbf{I}_{1D} \in \mathcal{T} \cup \mathcal{S} = \{\mathbf{T}^{(0)}, \ldots, \mathbf{T}^{(m)}, \mathbf{S}^{(0)}, \ldots \mathbf{S}^{(m)}\}$, we first compute its average along variable dimension as $\bar{\mathbf{I}}_{1D} \in \mathbb{R}^{D_m}$. Then, we apply Fast Fourier Transform (FFT) to extract its dominant frequencies with top K amplitudes:

$$\mathcal{F}, \mathcal{A} = \text{FFT}\left(\bar{\mathbf{I}}_{1D}\right), \tag{5}$$

where $\mathcal{F} = \{f_1, \ldots, f_K\}, \mathcal{A} = \{a_1, \ldots, a_K\}$ denotes denotes the amplitudes and their corresponding frequencies respectively. The corresponding periods are then computed as:

$$p_k = \lfloor \frac{D_m}{f_k} \rfloor, \tag{6}$$

where $k \in \{1, \ldots, K\}$. Finally, each $\mathbf{I}_{1D}$ is padded and reshaped into K 2D components driven by its dominant periods:

$$\begin{aligned} \mathbf{I}'^{(k)}_{1D} &= \text{Pad}\left(\mathbf{I}_{1D}, p_k\right), \\ \mathbf{I}^{(k)}_{2D} &= \text{Reshape}\left(\mathbf{I}'^{(k)}_{1D}, p_k\right), \end{aligned} \tag{7}$$

where $\mathbf{I}'^{(k)}_{1D} \in \mathbb{R}^{N \times D^{(k)}_m}, \mathbf{I}^{(k)}_{2D} \in \mathbb{R}^{N \times \frac{D^{(k)}_m}{p_k} \times p_k}$, and $D^{(k)}_m = \lceil \frac{D_m}{p_k} \rceil \cdot p_k$.

Convolution and Aggregation For each 1D component $\mathbf{I}_{1D} \in \mathcal{T} \cup \mathcal{S}$, we generate K corresponding 2D components $\{\mathbf{I}^{(1)}_{2D}, \ldots, \mathbf{I}^{(K)}_{2D}\}$, resulting in a total of $2(M+1) \cdot K$ representations. We then apply a shared 2D concolution backbone to each representation, to extract rich temporal features:

$$\hat{\mathbf{I}}^{(k)}_{2D} = \text{Conv2D}(\mathbf{I}^{(k)}_{2D}). \tag{8}$$

We adopt the Inception Block as the backbone due to its efficiency and effectiveness. It employs convolution kernels of varying sizes in parallel, which enables it to capture multi-scale features simultaneously with varying receptive fields. Next, each 2D representation is reshaped and unpadded back to its 1D form:

$$\begin{aligned} \hat{\mathbf{I}}'^{(k)}_{1D} &= \text{Reshape}\left(\hat{\mathbf{I}}^{(k)}_{2D}, p_k\right), \\ \hat{\mathbf{I}}^{(k)}_{1D} &= \text{Unpad}\left(\hat{\mathbf{I}}'^{(k)}_{1D}, p_k\right), \end{aligned} \tag{9}$$

where $\hat{\mathbf{I}}'^{(k)}_{1D} \in \mathbb{R}^{N \times D^{(k)}_m}, \hat{\mathbf{I}}^{(k)}_{1D} \in \mathbb{R}^{N \times D_m}$. Finally, we fuse all K 1D representations that derive from one 1D component into a single 1D representation via amplitude-softmax-weighted average:

$$\begin{aligned} \{w_1, \ldots, w_K\} &= \text{Softmax}(\{a_1, \ldots, a_K\}), \\ \hat{\mathbf{I}}_{1D} &= \sum_{k=1}^{K} w_k \cdot \hat{\mathbf{I}}^{(k)}_{1D}, \end{aligned} \tag{10}$$

where $\hat{\mathbf{I}}_{1D} \in \mathbb{R}^{N \times D_m}$. The final outputs are the transformed multi-scale trends $\hat{\mathcal{T}} = \{\hat{\mathbf{T}}^{(0)}, \ldots, \hat{\mathbf{T}}^{(M)}\}$ and seasons $\hat{\mathcal{S}} = \{\hat{\mathbf{S}}^{(0)}, \ldots, \hat{\mathbf{S}}^{(M)}\}$.

Multi-Scale Recursive Aggregation

To aggregate the multi-scale representation of trends $\hat{\mathcal{S}}$ and seasons $\hat{\mathcal{T}}$, we design a Multi-Scale Recursive Aggregation module that utilizes cross-attention to fuse representations recursively. Concretely, starting with the most global-scale representations $\mathbf{E}^{(M)}_{\mathbf{T}} = \hat{\mathbf{T}}^{(M)}$ and $\mathbf{E}^{(M)}_{\mathbf{S}} = \hat{\mathbf{S}}^{(M)}$, we recursively fuse them into their previous scales via cross-attention:

$$\begin{aligned} \mathbf{E}^{(m)}_{\mathbf{T}} &= \hat{\mathbf{T}}^{(m)} + \text{MHCA}\left(\hat{\mathbf{T}}^{(m)}, \mathbf{E}^{(m+1)}_{\mathbf{T}}, \mathbf{E}^{(m+1)}_{\mathbf{T}}\right), \\ \mathbf{E}^{(m)}_{\mathbf{S}} &= \hat{\mathbf{S}}^{(m)} + \text{MHCA}\left(\hat{\mathbf{S}}^{(m)}, \mathbf{E}^{(m+1)}_{\mathbf{S}}, \mathbf{E}^{(m+1)}_{\mathbf{S}}\right), \end{aligned} \tag{11}$$

where $m \in \{0, \ldots, M-1\}$, and $\mathbf{E}^{(m)}_{\mathbf{T}}, \mathbf{E}^{(m)}_{\mathbf{S}} \in \mathbb{R}^{N \times D_m}$. MHCA $(\cdot)$ denotes Multi-Head-Cross-Attention module, whose query matrix $\mathbf{Q}$, key matrix $\mathbf{K}$, and value matrix $\mathbf{V}$ are computed from its three arguments respectively. With this module, lower-resolution features are recursively fused into higher-resolution representations attentively, capturing both global and local patterns. Finally, the aggregated multi-scale representation of trend $\mathbf{E}^{(0)}_{\mathbf{T}}$ and season $\mathbf{E}^{(0)}_{\mathbf{S}}$ are added to obtain temporally enhanced fine-grained variable embeddings:

$$\mathbf{E}_1 = \mathbf{E}^{(0)}_{\mathbf{T}} + \mathbf{E}^{(0)}_{\mathbf{S}}, \tag{12}$$

where $\mathbf{E}_1 \in \mathbb{R}^{N \times D}$.

LLM Semantic Enhancement

Pretrained LLMs, such as the GPT series (Radford et al. 2018; Achiam et al. 2023), possess powerful language understanding capabilities and can generate highly compressed semantic representations, which we leverage to enhance the fine-grained variable embeddings $\mathbf{E}_1$.

Descriptive Prompt as Prefix In this study, we construct textual prompts based on task description and input statistics, as illustrated in Figure 2, and extract semantic embeddings with a frozen GPT-2 (Radford et al. 2019). Concretely, we first construct prompt $\mathbf{P}_i$ for all variables, then tokenize them:

$$\mathbf{H}_1 = \text{Tokenizer}\left(\{\mathbf{P}_1, \ldots, \mathbf{P}_N\}\right), \tag{13}$$

where $\mathbf{H}_1 \in \mathbb{R}^{N \times G}$, and G denotes the number of tokens. Next, we feed the output into the frozen GPT-2 encoder to obtain semantic embeddings:

$$\mathbf{H}_2 = \text{GPT-2}\left(\mathbf{H}_1\right), \tag{14}$$

where $\mathbf{H}_2 \in \mathbb{R}^{N \times G \times L}$, and L denotes GPT-2's embedding size. Finally, we average $\mathbf{H}_2$ along token dimension to obtain the semantic variable embeddings $\mathbf{E}_2 \in \mathbb{R}^{N \times L}$.

Cross-Modal Fusion In order to leverage the semantic variable embeddings $\mathbf{E}_2$, we fuse it with the temporal variable embeddings $\mathbf{E}_1$ via cross-attention, effectively enhancing the fine-grained representations with LLM. Specifically, we first permute $\mathbf{E}_1$ and $\mathbf{E}_2$ to $\mathbf{E}'_1 \in R^{D \times N}$ and $\mathbf{E}'_2 \in R^{L \times N}$, regarding them as D and L tokens respectively. Then, we apply a MHCA $(\cdot)$ to align the two modalities:

$$\mathbf{E}'_3 = \mathbf{E}'_1 + \text{MHCA}\left(\mathbf{E}'_1, \mathbf{E}'_2, \mathbf{E}'_2\right), \tag{15}$$

where $\mathbf{E}'_3 \in \mathbb{R}^{D \times N}$. Finally, the output is permuted back to $\mathbf{E}_3 \in \mathbb{R}^{N \times D}$, which is the LLM-enhanced variable embeddings.

Output Projection

The enhanced variable embeddings $\mathbf{E}_3$ encodes rich temporal and semantic information, thus we introduce it as a residual term to the original variable embeddings $\mathbf{E}_0$ to generate the final variable embeddings:

$$\mathbf{E}_4 = \mathbf{E}_0 + \mathbf{E}_3. \tag{16}$$

Finally, we project $\mathbf{E}_4$ to forecasting results by linear transformation:

$$\hat{\mathbf{Y}} = \text{Linear}\left(\mathbf{E}_4\right), \tag{17}$$

where $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times F}$ denotes the predicted values of all variables over the next F time steps.

Adaptive Mixed Loss To enhance the robustness and generalization of the model under varying error distributions, we introduce an Adaptive Mixed Loss. It dynamically adjusts the weights between L1 and L2 branches based on the magnitude of prediction errors:

$$\begin{aligned} \mathscr{L} &= \text{Avg}\left(\alpha \odot |\hat{\mathbf{Y}} - \mathbf{Y}| + (1-\alpha) \odot ||\hat{\mathbf{Y}} - \mathbf{Y}||_2\right), \\ \alpha &= \text{Sigmoid}\left(\gamma \cdot |\hat{\mathbf{Y}} - \mathbf{Y}|\right), \end{aligned} \tag{18}$$

where γ is a learnable parameter , and $\odot$ denotes element-wise multiplication. This approach makes the loss more robust to outliers, resulting in more stable training.

Experiments

Baselines and Datasets

To evaluate the effectiveness of M3Time, we conduct extensive experiments of different tasks on seven public real-world datasets across varying domains: ETTh1, ETTh2, ETTm1, ETTm2 (Zhou et al. 2021), Weather (Rasp et al. 2020), ECL (Gasparin, Lukovic, and Alippi 2019), and Traffic (Jiang et al. 2021), covering diverse sampling frequencies, number of variables, and temporal patterns. We compare M3Time with 10 strong baselines, including TimeCMA (Liu et al. 2025), Time-LLM (Jin et al. 2023a), TimerMixer++ (Wang et al. 2024a), TimerMixer (Wang et al. 2024b), PatchTST (Nie et al. 2022), FEDformer (Zhou et al. 2022), TiDE (Das et al. 2023b), TimesNet (Wu et al. 2022), DLinear (Zeng et al. 2023), and more.

Main Results

Long-term Forecasting In long-term forecasting tasks, models need to capture and predict time series trends over extended horizons, which is critically important for applications such as weather forecasting, traffic planning, and energy management. To thoroughly evaluate the model's long-term forecasting performance across various domains, we conduct experiments on seven representative real-world datasets: ETTh1, ETTh2, ETTm1, ETTm2, Weather, ECL, and Traffic. Table 1 demonstrates that M3Time consistently outperforms other models in long-term forecasting task. Compared with large language model–based baselines, M3Time also demonstrates clear advantages. Against TimeCMA and Time-LLM, M3Time consistently delivers better performance across all datasets. In addition, compared with the second-best model TimeMixer++, M3Time achieves significant improvements on all four ETT datasets, with an average reduction of 4.6% in MSE and 2.4% in MAE. Additionally, on the ECL dataset, M3Time reduces MSE by over 3%, and on the Traffic dataset, it achieves a 3.6% improvement in MSE. These results highlight the domain-agnostic effectiveness of M3Time in long-term forecasting task.

Few-shot Forecasting In real-world applications, time series forecasting models often face the challenge of limited training data. To evaluate the models' transferability and adaptability under few-shot conditions, we conduct experiments on six diverse datasets, using only 10% of each dataset. This experimental setup is designed to test the models' adaptability to sparse data and their ability to capture generalizable patterns, highlighting their practical value in real-world scenarios with data scarcity. Table 2 shows that among all competing methods, M3Time achieves the best performance on five datasets. Compared to the second-best model TimeMixer++, M3Time reduces the average MSE by 5.1% and MAE by 3.7%.

Zero-shot Forecasting To evaluate the model's generalization capability, we train the model on a source dataset, then test it directly on an unseen target dataset without any additional fine-tuning. This cross-domain general setup assesses the model's adaptability and predictive robustness

Methods	**M3Time**		TimeCMA		TimeLLM		TimeMixer++		TimeMixer		iTransformer		PatchTST		TimesNet		DLinear		FEDformer		Autoformer	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.398**	**0.417**	0.423	0.431	0.448	0.443	0.419	0.432	0.447	0.440	0.454	0.447	0.516	0.484	0.458	0.450	0.461	0.457	0.498	0.484	0.496	0.487
ETTh2	**0.329**	**0.376**	0.372	0.397	0.381	0.404	0.339	0.380	0.364	0.395	0.383	0.407	0.391	0.411	0.414	0.427	0.563	0.519	0.437	0.449	0.450	0.459
ETTm1	**0.352**	**0.372**	0.380	0.392	0.410	0.409	0.369	0.378	0.381	0.395	0.407	0.410	0.406	0.407	0.400	0.406	0.404	0.408	0.448	0.452	0.588	0.517
ETTm2	**0.254**	**0.309**	0.275	0.323	0.296	0.340	0.269	0.320	0.275	0.323	0.288	0.332	0.290	0.334	0.291	0.333	0.354	0.402	0.305	0.349	0.327	0.371
Weather	0.227	**0.260**	0.250	0.276	0.275	0.291	**0.226**	0.262	0.240	0.271	0.258	0.278	0.265	0.285	0.259	0.287	0.265	0.315	0.309	0.360	0.338	0.382
ECL	**0.160**	0.254	0.174	0.269	0.195	0.288	0.165	**0.253**	0.182	0.272	0.178	0.270	0.216	0.318	0.192	0.304	0.225	0.319	0.214	0.327	0.227	0.338
Traffic	**0.401**	0.266	–	–	0.415	0.279	0.416	**0.264**	0.484	0.297	0.428	0.282	0.529	0.341	0.667	0.426	0.625	0.383	0.610	0.376	0.628	0.379

Table 1: Long-term forecasting results. All results are averaged from four different forecasting horizons: $\{96, 192, 336, 720\}$. A lower value indicates better performance. Our full results are in Appendix.

Methods	**M3Time**		TimeMixer++		TimeMixer		iTransformer		TiDE		Crossformer		DLinear		PatchTST		TimesNet		FEDformer		Autoformer	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.468**	**0.471**	0.517	0.512	0.613	0.520	0.510	0.597	0.589	0.535	0.645	0.558	0.691	0.600	0.633	0.542	0.869	0.628	0.639	0.561	0.702	0.596
ETTh2	**0.360**	0.402	0.379	**0.391**	0.402	0.433	0.455	0.461	0.395	0.412	0.645	0.447	0.605	0.538	0.415	0.431	0.479	0.465	0.466	0.475	0.488	0.499
ETTm1	**0.366**	**0.384**	0.398	0.431	0.487	0.461	0.491	0.516	0.425	0.458	0.462	0.489	0.411	0.429	0.501	0.466	0.677	0.537	0.722	0.605	0.802	0.628
ETTm2	**0.258**	**0.315**	0.291	0.351	0.311	0.367	0.375	0.412	0.317	0.371	0.343	0.389	0.316	0.368	0.296	0.343	0.320	0.353	0.463	0.488	1.342	0.930
Weather	**0.236**	**0.271**	0.241	**0.271**	0.242	0.281	0.291	0.331	0.249	0.291	0.267	0.306	0.241	0.283	0.324	0.279	0.279	0.301	0.284	0.324	0.300	0.342
ECL	0.200	0.300	**0.168**	**0.271**	0.187	0.277	0.241	0.337	0.196	0.289	0.214	0.308	0.180	0.280	0.180	0.273	0.323	0.392	0.346	0.427	0.431	0.478

Table 2: Few-shot learning on 10% training data. All results are averaged from four different forecasting horizons: $H \in \{96, 192, 336, 720\}$.

when facing different data distributions and feature patterns. Table 3 reports the zero-shot prediction results under four cross-domain zero-shot forecasting settings, demonstrating the advantage of M3Time in zero-shot generalization. Notably, compared with the strong baseline TimeMixer++, M3Time achieves a reduction of 4.7% in MSE and 3.4% in MAE. In addition, relative to iTransformer, M3Time reduces MSE by 20.7% and MAE by 15.9%. These results demonstrate the robustness and strong generalization capability, underscoring its potential for real-world applications.

Ablation Studies

Model Design We conduct ablation studies on four ETT series datasets with prediction length set to 720, to evaluate the effectiveness of each component of M3Time. As shown in Table 4, the complete M3Time, with all components included, achieves the best result with an average MSE of 0.395. Specifically, removing the Cross-Variable Interaction module results in a 2.71% increase in MSE, indicating its critical role in capturing inter-variable dependencies; excluding the Trend-Seasonal Decomposition module yields a 3.89% improvement, confirming the importance of disentangling trend and season for time series modeling; removing the Dynamic Convolution Block leads to a 4.13% improvement, highlighting its contribution to extracting fine-grained representations; finally, eliminating the LLM Semantic Enhancement module renders a 3.42% increase in MSE, demonstrating its effectiveness in incorporating semantic priors to improve numerical forecasting.

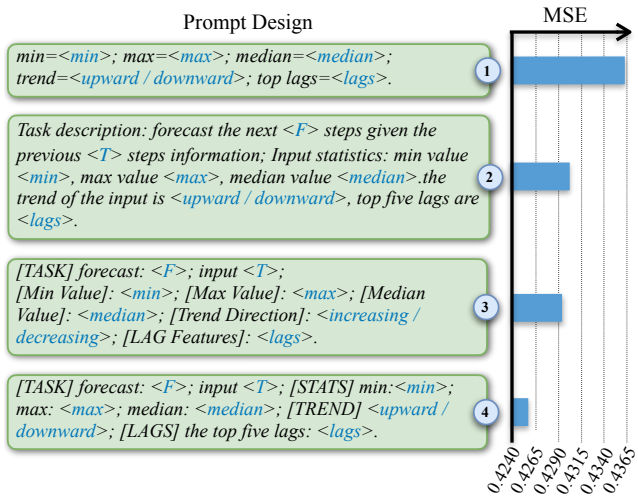

Figure 4: Four prompt design choices and their corresponding forecasting results. <> denotes task-specific parameters or calculated input statistics.

Prompt Design To gain insight into the form of textual prompts that can effectively guide forecasting, we design four semantically similar but structurally different prompts and evaluate their corresponding forecasting MSE on the ETTh1 dataset with prediction length set to 720. As shown in Figure 4, the first prompt presents input statistics as simple key-value pairs; the second adopts a narrative format;

Methods	**M3Time**		TimeMixer++		TimeMixer		LLMTime		DLinear		PatchTST		TimesNet		iTransformer		Fedformer		Autoformer	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1→ ETTh2	**0.322**	**0.372**	0.367	0.391	0.427	0.424	0.992	0.708	0.493	0.488	0.380	0.405	0.421	0.431	0.481	0.474	0.712	0.693	0.634	0.651
ETTh2 → ETTh1	**0.489**	**0.484**	0.511	0.498	0.679	0.577	1.961	0.981	0.703	0.574	0.565	0.513	0.865	0.621	0.552	0.511	0.612	0.624	0.599	0.571
ETTm1 → ETTm2	**0.270**	**0.322**	0.291	0.331	0.329	0.357	1.867	0.869	0.335	0.389	0.296	0.334	0.322	0.354	0.324	0.331	0.612	0.611	0.603	0.592
ETTm2 → ETTm1	0.438	**0.435**	**0.427**	0.448	0.554	0.478	1.933	0.984	0.649	0.537	0.568	0.492	0.769	0.567	0.559	0.491	0.577	0.601	0.594	0.597

Table 3: Zero-shot learning results. All results are averaged from four different forecasting horizons: $H \in \{96, 192, 336, 720\}$.

	ETTh1	ETTh2	ETTm1	ETTm2	Average
M3Time	**0.423**	**0.387**	**0.418**	**0.353**	**0.395**
w/o Cross-Variable Interaction	0.444	0.392	0.423	0.363	0.406
w/o Trend-Seasonal Decomposition	0.438	0.408	0.433	0.365	0.411
w/o Dynamic Convolution Block	0.446	0.397	0.434	0.372	0.412
w/o LLM Semantic Enhancement	0.431	0.395	0.421	0.390	0.409

Table 4: MSE for long-term forecasting across 4 benchmarks, evaluated with different model components. Only results for the prediction length of 720 are reported in the table.

the third adopts a semi-structured slot style with bracketed tags for each field; the fourth further formalizes the structure by organizing fields under explicit headers. Despite being semantically equivalent, their corresponding forecasting performance decreases monotonically from the first to the fourth. We attribute this trend to the increasing structural clarity and tokenization consistency offered by more canonical schemas. In particular, the headered format provides stable anchor points for cross-modal alignment, enabling more effective integration of semantic and numerical information. In contrast, the key-value and narrative forms introduce linguistic variability and ambiguous token boundaries, which can hinder alignment and reduce the utility of textual priors in numerical prediction.

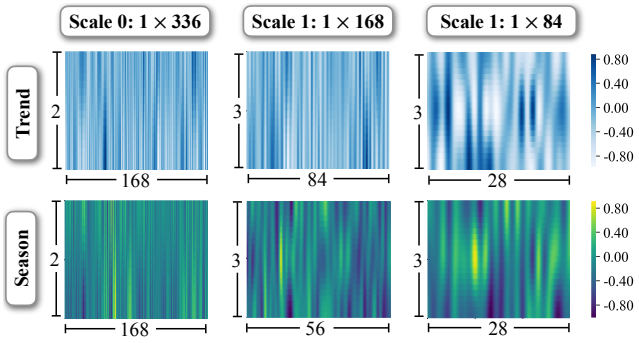

Figure 5: Visualization of a sample sequence from ETTh1 after being decomposed, downsampled, then 2D transformed.

Visual Analytics As shown in Figure 5, the trend and season components exhibit distinct periodicity across scales. Specifically, as the resolution decreases, the energy distribution of the trend component progressively concentrates in lower-frequency regions. In contrast, the energy distribution of the season component shifts toward higher-frequency regions as the resolution increases. In other words, it demonstrates that M3Time can simultaneously capture scale-dependent trends and seasonal patterns effectively. As shown in Figure 6, the similarity matrix before Cross-Modal Fusion exhibits moderate correlations. However, after Cross-Modal Fusion, certain off-diagonal similarities are significantly reduced, resulting in a sparser and more structured similarity matrix. This indicates that M3Time can selectively strength or weaken representations through the guidance of textual prompts, thus retain useful features or suppress useless noise, which improves the overall forecasting performance.

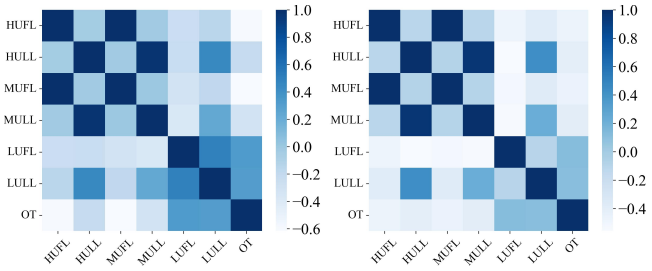

Figure 6: Comparison of the cosine similarity heatmaps of variable embeddings before and after Cross-Modal Fusion.

Conclusion

In this study, we revisit the role of textual priors in MTSF and argue that only strong base models that extract fine-grained numerical representations can genuinely benefit from LLM semantics. Building on these insights, we present M3Time, a LLM-enhanced, multi-modal, multi-scale, and multi-frequency framework for multivariate time series forecasting. Furthermore, we conduct extensive experiments on seven real-world datasets, showcasing M3Time's state-of-the-art long-term forecasting performance, few-shot transferability, and zero-shot generalization capability. We also conduct thorough ablation studies to verify the effectiveness of all modules and gain insights into the underlying mechanisms.

Acknowledgments

This work was supported by the National Key R&D Program of China (2022YFB4701400/4701402), SSTIC Grant (KJZD20230923115106012, KJZD20230923114916032, GJHZ20240218113604008).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ansari, A. F.; Stella, L.; Turkmen, C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S. S.; Arango, S. P.; Kapoor, S.; et al. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.

Bi, K.; Xie, L.; Zhang, H.; Chen, X.; Gu, X.; and Tian, Q. 2023. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619(7970): 533–538.

Bumb, M.; Vemulapalli, A.; Jella, S. H. V. P.; Gupta, A.; La, A.; Rossi, R. A.; Chen, H.; Dernoncourt, F.; Ahmed, N. K.; and Wang, Y. 2025. Forecasting Time Series with LLMs via Patch-Based Prompting and Decomposition. *arXiv preprint arXiv:2506.12953*.

Cao, D.; Jia, F.; Arik, S. O.; Pfister, T.; Zheng, Y.; Ye, W.; and Liu, Y. 2023. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*.

Challu, C.; Olivares, K. G.; Oreshkin, B. N.; Ramirez, F. G.; Canseco, M. M.; and Dubrawski, A. 2023. NHITS: neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 6989–6997.

Das, A.; Kong, W.; Leach, A.; Mathur, S.; Sen, R.; and Yu, R. 2023b. Long-term forecasting with tide: Time-series dense encoder. arXiv 2023. *arXiv preprint arXiv:2304.08424*.

Das, A.; Kong, W.; Sen, R.; and Zhou, Y. 2024. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*.

Franceschi, J.-Y.; Dieuleveut, A.; and Jaggi, M. 2019. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32.

Gasparin, A.; Lukovic, S.; and Alippi, C. 2019. Deep Learning for Time Series Forecasting: The Electric Load Case. *arXiv preprint arXiv:1907.09207*. Electricity load forecasting dataset (ECL and related).

Goswami, M.; Szafer, K.; Choudhry, A.; Cai, Y.; Li, S.; and Dubrawski, A. 2024. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*.

Gruver, N.; Finzi, M.; Qiu, S.; and Wilson, A. G. 2023. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36: 19622–19635.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Jalalifar, R.; Delavar, M. R.; and Ghaderi, S. F. 2024. SAC-ConvLSTM: A novel spatio-temporal deep learning-based approach for a short term power load forecasting. *Expert Systems with Applications*, 237: 121487.

Jia, F.; Wang, K.; Zheng, Y.; Cao, D.; and Liu, Y. 2024. Gpt4mts: Prompt-based large language model for multimodal time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 23343–23351.

Jiang, R.; Yin, D.; Wang, Z.; Wang, Y.; Deng, J.; Liu, H.; Cai, Z.; Deng, J.; Song, X.; and Shibasaki, R. 2021. DL-Traff: Survey and Benchmark of Deep Learning Models for Urban Traffic Prediction. In *arXiv preprint arXiv:2108.09091*. Traffic forecasting datasets benchmark (METR-LA, PeMS, etc.).

Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J. Y.; Shi, X.; Chen, P.-Y.; Liang, Y.; Li, Y.-F.; Pan, S.; et al. 2023a. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*.

Jin, M.; Wen, Q.; Liang, Y.; Zhang, C.; Xue, S.; Wang, X.; Zhang, J.; Wang, Y.; Chen, H.; Li, X.; et al. 2023b. Large models for time series and spatio-temporal data: A survey and outlook. *arXiv preprint arXiv:2310.10196*.

Li, Z.; Lin, X.; Liu, Z.; Zou, J.; Wu, Z.; Zheng, L.; Fu, D.; Zhu, Y.; Hamann, H.; Tong, H.; et al. 2025. Language in the flow of time: Time-series-paired texts weaved into a unified temporal narrative. *arXiv preprint arXiv:2502.08942*.

Liu, C.; Xu, Q.; Miao, H.; Yang, S.; Zhang, L.; Long, C.; Li, Z.; and Zhao, R. 2025. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 18780–18788.

Liu, C.; Yang, S.; Xu, Q.; Li, Z.; Long, C.; Li, Z.; and Zhao, R. 2024a. Spatial-temporal large language model for traffic prediction. In *2024 25th IEEE International Conference on Mobile Data Management (MDM)*, 31–40. IEEE.

Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; and Xu, Q. 2022. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35: 5816–5828.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.

Liu, Y.; Qin, G.; Huang, X.; Wang, J.; and Long, M. 2024b. AutoTimes: Autoregressive Time Series Forecasters via Large Language Models. In *Advances in Neural Information Processing Systems*, volume 37, 122154–122184. Curran Associates, Inc.

Nematirad, R.; Pahwa, A.; and Natarajan, B. 2025. Times2d: Multi-period decomposition and derivative mapping for general time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 19651–19658.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. *OpenAI Preprint*. ArXiv:1801.06146.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI*.

Rasp, S.; Dueben, P. D.; Scher, S.; Weyn, J. A.; Mouatadid, S.; and Thuerey, N. 2020. WeatherBench: A Benchmark Dataset for Data-Driven Weather Forecasting. *arXiv preprint arXiv:2002.00469*. Benchmark dataset derived from ERA5 reanalysis data for deep learning forecasting models.

Rasul, K.; Ashok, A.; Williams, A. R.; Khorasani, A.; Adamopoulos, G.; Bhagwatkar, R.; Biloš, M.; Ghonia, H.; Hassen, N.; Schneider, A.; et al. 2023. Lag-llama: Towards foundation models for time series forecasting. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*.

Salinas, D.; Flunkert, V.; Gasthaus, J.; and Januschowski, T. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3): 1181–1191.

Shen, L.; Li, Z.; and Kwok, J. 2020. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in neural information processing systems*, 33: 13016–13026.

Wang, H.; Peng, J.; Huang, F.; Wang, J.; Chen, J.; and Xiao, Y. 2023. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*.

Wang, S.; Li, J.; Shi, X.; Ye, Z.; Mo, B.; Lin, W.; Ju, S.; Chu, Z.; and Jin, M. 2024a. Timemixer++: A general time series pattern machine for universal predictive analysis. *arXiv preprint arXiv:2410.16032*.

Wang, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J. Y.; and Zhou, J. 2024b. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*.

Woo, G.; Liu, C.; Kumar, A.; Xiong, C.; Savarese, S.; and Sahoo, D. 2024. Unified Training of Universal Time Series Forecasting Transformers. arXiv:2402.02592.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34: 22419–22430.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11106–11115.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, 27268–27286. PMLR.

Zhou, T.; Niu, P.; Wang, X.; Sun, L.; and Jin, R. 2023. One Fits All: Power General Time Series Analysis by Pretrained LM. *Advances in Neural Information Processing Systems*, 36.